PEMROGRAMAN BERBASIS OBJEK

Created by Aris Prawisudatama

HELLO THERE

I am

ARIS PRAWISUDATAMA

A Software Developer.

GITHUB REPOSITORY

https://github.com/soedomoto/usm_tis21213p

GITHUB REPOSITORY

ttps://github.com/soedomoto/usm_tis21213

KENALI AUDIENCE

Kapan Anda pertama kali membangun aplikasi (web/mobile/desktop) ?
Aplikasi apakah itu?
Teknologi apa saja yang digunakan?

REAL WORLD FACTS

Gunakan version control
Gunakan framework

Gunakan multi env: development, staging, production
Gunakan DB migration dan seeder

REAL WORLD FACTS

Gunakan version control

Gunakan framework

Gunakan multi env: development, staging, production

Gunakan DB migration dan seeder

REAL WORLD FACTS

Gunakan version control

Gunakan framework

Gunakan multi env: development, staging, production

Gunakan DB migration dan seeder

REAL WORLD FACTS

Gunakan version control
Gunakan framework

Cepat. Fokus ke business, tidak perlu memaintain dependency

Gunakan multi env: development, staging, production
Gunakan DB migration dan seeder

REAL WORLD FACTS

Gunakan version control
Gunakan framework

Cepat. Fokus ke business, tidak perlu memaintain dependency

kan multi env: development, staging, produ

Gunakan DB migration dan seeder

REAL WORLD FACTS

Gunakan version control
Gunakan framework

Cepat. Fokus ke business, tidak perlu memaintain dependency

kan multi env: development, staging, produ
Gunakan DB migration dan seeder

PARADIGMA PEMROGRAMAN

Pemrograman Prosedural
Pemrograman Berbasis Objek

> *“OO makes code understandable by encapsulating moving parts. FP makes code understandable by minimizing moving parts.”*

PARADIGMA PEMROGRAMAN

Pemrograman Prosedural

Pemrograman Berbasis Objek

“OO makes code understandable by encapsulating moving parts. FP makes code understandable by minimizing moving parts.”

PARADIGMA PEMROGRAMAN

Pemrograman Prosedural

Pemrograman Berbasis Objek

“OO makes code understandable by encapsulating moving parts. FP makes code understandable by minimizing moving parts.”

PEMROGRAMAN PROSEDURAL

Memiliki algoritma pemecahan masalah yang sederhana, standar dan efektif
Penulisan program memiliki struktur logika yang benar dan mudah dipahami
Program hanya terdiri dari 3(tiga) struktur dasar, yaitu struktur berurutan, struktur seleksi dan struktur perulangan

PEMROGRAMAN PROSEDURAL

Memiliki algoritma pemecahan masalah yang sederhana, standar dan efektif
Penulisan program memiliki struktur logika yang benar dan mudah dipahami
Program hanya terdiri dari 3(tiga) struktur dasar, yaitu struktur berurutan, struktur seleksi dan struktur perulangan

PEMROGRAMAN PROSEDURAL

Memiliki algoritma pemecahan masalah yang sederhana, standar dan efektif

Penulisan program memiliki struktur logika yang benar dan mudah dipahami

Program hanya terdiri dari 3(tiga) struktur dasar, yaitu struktur berurutan, struktur seleksi dan struktur perulangan

PEMROGRAMAN PROSEDURAL

Memiliki algoritma pemecahan masalah yang sederhana, standar dan efektif

Penulisan program memiliki struktur logika yang benar dan mudah dipahami

Program hanya terdiri dari 3(tiga) struktur dasar, yaitu struktur berurutan, struktur seleksi dan struktur perulangan

KEKURANGAN PEMROGRAMAN PROSEDURAL

Program cukup sulit untuk proses perawatan
Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan
Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models
Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain

KEKURANGAN PEMROGRAMAN PROSEDURAL

Program cukup sulit untuk proses perawatan
Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan
Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models
Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain

KEKURANGAN PEMROGRAMAN PROSEDURAL

Program cukup sulit untuk proses perawatan
Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan
Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models
Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain

KEKURANGAN PEMROGRAMAN PROSEDURAL

Program cukup sulit untuk proses perawatan
Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan
Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models
Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain

KEKURANGAN PEMROGRAMAN PROSEDURAL

Program cukup sulit untuk proses perawatan
Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan
Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models
Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain

PEMROGRAMAN BERBASIS OBJEK

Dengan OOP, kode-kode yang kita buat menjadi lebih rapih dan terstruktur
Dengan OOP, proses reuse kode-kode yang kita buat untuk project yang hampir sama, mudah, karena kode yang kita buat rapih dan terstruktur. Sehingga untuk mengubah atau reuse kode jauh lebih mudah

PEMROGRAMAN BERBASIS OBJEK

Dengan OOP, kode-kode yang kita buat menjadi lebih rapih dan terstruktur

Dengan OOP, proses reuse kode-kode yang kita buat untuk project yang hampir sama, mudah, karena kode yang kita buat rapih dan terstruktur. Sehingga untuk mengubah atau reuse kode jauh lebih mudah

PEMROGRAMAN BERBASIS OBJEK

Dengan OOP, kode-kode yang kita buat menjadi lebih rapih dan terstruktur

Dengan OOP, proses reuse kode-kode yang kita buat untuk project yang hampir sama, mudah, karena kode yang kita buat rapih dan terstruktur. Sehingga untuk mengubah atau reuse kode jauh lebih mudah

PEMROGRAMAN BERBASIS OBJEK

Konsepnya per modul. Fungsi atau prosedur misal di java atau di PHP tinggal panggil saja nama fungsinya. Jika bermasalah nantinya akan mudah diketahui karena terbaca dari fungsi yang kita panggil. Itulah yang dimaksud konsep per modul. Sehingga dengan OOP, kita dimudahkan untuk membuat dan membaca kode kita (efisiensi kode)

Konsep OOP juga memudahkan kita untuk menganalisa program yang kita akan buat. Ini akan sangat terasa kalau kita membuat program besar dan rumit

PEMROGRAMAN BERBASIS OBJEK

Konsepnya per modul. Fungsi atau prosedur misal di java atau di PHP tinggal panggil saja nama fungsinya. Jika bermasalah nantinya akan mudah diketahui karena terbaca dari fungsi yang kita panggil. Itulah yang dimaksud konsep per modul. Sehingga dengan OOP, kita dimudahkan untuk membuat dan membaca kode kita (efisiensi kode)

Konsep OOP juga memudahkan kita untuk menganalisa program yang kita akan buat. Ini akan sangat terasa kalau kita membuat program besar dan rumit

PEMROGRAMAN BERBASIS OBJEK

Konsepnya per modul. Fungsi atau prosedur misal di java atau di PHP tinggal panggil saja nama fungsinya. Jika bermasalah nantinya akan mudah diketahui karena terbaca dari fungsi yang kita panggil. Itulah yang dimaksud konsep per modul. Sehingga dengan OOP, kita dimudahkan untuk membuat dan membaca kode kita (efisiensi kode)

Konsep OOP juga memudahkan kita untuk menganalisa program yang kita akan buat. Ini akan sangat terasa kalau kita membuat program besar dan rumit

KEKURANGAN PEMROGRAMAN BERBASIS OBJEK

Kesulitan dalam ide programnya yang dapat digunakan dalam program

Membangun awal perlu ketrampilan programing lebih

KEKURANGAN PEMROGRAMAN BERBASIS OBJEK

Kesulitan dalam ide programnya yang dapat digunakan dalam program

Membangun awal perlu ketrampilan programing lebih

KEKURANGAN PEMROGRAMAN BERBASIS OBJEK

Kesulitan dalam ide programnya yang dapat digunakan dalam program

Membangun awal perlu ketrampilan programing lebih

PREREQUISITES

Apache/Nginx + PHP (XAMPP, etc)
Editor dengan Code Highlighting (VSCode, Sublime, Notepad++)

PREREQUISITES

Apache/Nginx + PHP (XAMPP, etc)
Editor dengan Code Highlighting (VSCode, Sublime, Notepad++)

PREREQUISITES

Apache/Nginx + PHP (XAMPP, etc)
Editor dengan Code Highlighting (VSCode, Sublime, Notepad++)

OBJECT DAN CLASS

Object secara umum dapat didefinisikan sebagai individu yang berdiri sendiri.

OBJECT DAN CLASS

Object secara umum dapat didefinisikan sebagai individu yang berdiri sendiri.

OBJECT DAN CLASS

Object secara umum dapat didefinisikan sebagai individu yang berdiri sendiri.

Contoh: Si Andi, Si Ahmad, Si Blacky, Si Garfield, Honda Beat Plat B1234AA

OBJECT DAN CLASS

Object secara umum dapat didefinisikan sebagai individu yang berdiri sendiri.

Contoh: Si Andi, Si Ahmad, Si Blacky, Si Garfield, Honda Beat Plat B1234AA

Sebuah object merupakan instance dari class

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

Si Andi mempunyai tipe(kelas) "Manusia"

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

Si Andi mempunyai tipe(kelas) "Manusia"

Si Blacky mempunyai tipe(kelas) "Anjing"

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

Si Andi mempunyai tipe(kelas) "Manusia"

Si Blacky mempunyai tipe(kelas) "Anjing"

Si Garfield mempunyai tipe(kelas) "Kucing"

OBJECT DAN CLASS

Sementara Class dapat didefinisikan sebagai tipe dari object tersebut

Si Andi mempunyai tipe(kelas) "Manusia"

Si Blacky mempunyai tipe(kelas) "Anjing"

Si Garfield mempunyai tipe(kelas) "Kucing"

Si Honda Beat Plat B1234AA mempunyai tipe(kelas) "Sepeda Motor"

OBJECT DAN CLASS DI PHP

```php
class Manusia {
  // Properties
  // Methods
}

$andi = new Manusia();
echo "andi adalah sebuah " . gettype($andi) . " bertipe " .
```

$andi adalah sebuah object yang mempunyai tipe/class Manusia

ATTRIBUTE ATAU PROPERTI

Setiap class dapat memiliki beberapa attribut

ATTRIBUTE ATAU PROPERTI

Setiap class dapat memiliki beberapa attribut

ATTRIBUTE ATAU PROPERTI

Setiap class dapat memiliki beberapa attribut

Sebagai instance dari class, object pasti memiliki attribut yang sama dengan kelasnya, tetapi masing-masing object dapat memiliki nilai yang berbeda

ATTRIBUTE ATAU PROPERTI

Setiap class dapat memiliki beberapa attribut

Sebagai instance dari class, object pasti memiliki attribut yang sama dengan kelasnya, tetapi masing-masing object dapat memiliki nilai yang berbeda

Si Blacky adalah "Anjing" berkaki 4

ATTRIBUTE ATAU PROPERTI

Setiap class dapat memiliki beberapa attribut

Sebagai instance dari class, object pasti memiliki attribut yang sama dengan kelasnya, tetapi masing-masing object dapat memiliki nilai yang berbeda

Si Blacky adalah "Anjing" berkaki 4

Si Doggy adalah "Anjing" berkaki 3

METHOD ATAU PERILAKU

Setiap class juga dapat memiliki beberapa method/perilaku

METHOD ATAU PERILAKU

Setiap class juga dapat memiliki beberapa method/perilaku

METHOD ATAU PERILAKU

Setiap class juga dapat memiliki beberapa method/perilaku

Sebagai instance dari class, object pasti memiliki method yang sama dengan kelasnya

METHOD ATAU PERILAKU

Setiap class juga dapat memiliki beberapa method/perilaku

Sebagai instance dari class, object pasti memiliki method yang sama dengan kelasnya

"Anjing" memiliki perilaku berjalan, menggonggong, menoleh, melet, buang air, dll

METHOD ATAU PERILAKU

Setiap class juga dapat memiliki beberapa method/perilaku

Sebagai instance dari class, object pasti memiliki method yang sama dengan kelasnya

"Anjing" memiliki perilaku berjalan, menggonggong, menoleh, melet, buang air, dll

Si Doggy sebagai seekor "Anjing" akan memiliki perilaku yang sama

ATTRIBUTE DAN METHOD DI PHP

Identifier public/protected/private akan dijelaskan di bagian Sifat OOP

```php
class Manusia {
  public $jumlahMata;
  public $jumlahTangan;
  public $jumlahKaki;
  private $golonganDarah;
  private $jenisKelamin;

  // Constructor dapat berisi nilai awal (default) ketika s
  public function __construct() {
    $this->jumlahMata = 2;
    $this->jumlahTangan = 2;
    $this->jumlahKaki = 2;
    $this->golonganDarah = "A";
    $this->jenisKelamin = "L";
  }
```

$jumlahMata adalah sebuah attribute.

golonganDarah() adalah sebuah method.

SIFAT OBJECT ORIENTED PROGRAMMING

Enkapsulasi (Encapsulation)

Pewarisan (Inheritance)

Polimorfisme (Polimorphism)

SIFAT OBJECT ORIENTED PROGRAMMING

Enkapsulasi (Encapsulation)
Pewarisan (Inheritance)
Polimorfisme (Polimorphism)

SIFAT OBJECT ORIENTED PROGRAMMING

Enkapsulasi (Encapsulation)
Pewarisan (Inheritance)
Polimorfisme (Polimorphism)

SIFAT OBJECT ORIENTED PROGRAMMING

Enkapsulasi (Encapsulation)
Pewarisan (Inheritance)
Polimorfisme (Polimorphism)

ENKAPSULASI

Singkatnya Enkapsulasi adalah sebuah proses penggabungan atribut dan method ke dalam sebuah kelas

ENKAPSULASI

Singkatnya Enkapsulasi adalah sebuah proses penggabungan atribut dan method ke dalam sebuah kelas

Tujuan dari enkapsulasi adalah membatasi bagaimana suatu atribut maupun method dapat digunakan, dengan menentukan hak akses (modifier)

ENKAPSULASI

Singkatnya Enkapsulasi adalah sebuah proses penggabungan atribut dan method ke dalam sebuah kelas

Tujuan dari enkapsulasi adalah membatasi bagaimana suatu atribut maupun method dapat digunakan, dengan menentukan hak akses (modifier)

Modifier terdiri dari: public, protected, private

ENKAPSULASI DI PHP

```php
class Manusia {
  public $jumlahMata;
  public $jumlahTangan;
  public $jumlahKaki;
  private $golonganDarah;
  private $jenisKelamin;

  // Constructor dapat berisi nilai awal (default) ketika s
  public function __construct() {
    $this->jumlahMata = 2;
    $this->jumlahTangan = 2;
    $this->jumlahKaki = 2;
    $this->golonganDarah = "A";
    $this->jenisKelamin = "L";
  }
```

$jumlahMata adalah sebuah **public attribute**, sehingga bisa diakses secara langsung.

$golonganDarah adalah sebuah **private attribute**, sehingga tidak bisa diakses secara langsung. Perlu disertakan sebuah **public method golonganDarah()** untuk mengaksesnya.

INHERITANCE

Singkatnya Inheritance adalah pewarisan sifat dari kelas induk

INHERITANCE

Singkatnya Inheritance adalah pewarisan sifat dari kelas induk
Yang diwariskan adalah semua attribute dan method yang bersifat **public** dan **protected**

INHERITANCE

Singkatnya Inheritance adalah pewarisan sifat dari kelas induk
Yang diwariskan adalah semua attribute dan method yang bersifat **public** dan **protected**
di PHP menggunakan keyword **extends**

INHERITANCE DI PHP

```php
class Manusia
{
  public $jumlahMata;
  public $jumlahTangan;
  public $jumlahKaki;
  private $golonganDarah;
  protected $jenisKelamin;

  // Constructor dapat berisi nilai awal (default) ketika s
  public function __construct()
  {
    $this->jumlahMata = 2;
    $this->jumlahTangan = 2;
    $this->jumlahKaki = 2;
    $this->golonganDarah = "A";
```

TukangOjek mewarisi semua sifat (attribut) dan perilaku (method) yang bersifat public dan protected dari induknya Manusia

POLIMORFISME

Singkatnya Polimorfisme adalah suatu perilaku (method) yang sama dapat memiliki banyak bentuk

POLIMORFISME

Singkatnya Polimorfisme adalah suatu perilaku (method) yang sama dapat memiliki banyak bentuk Misalnya "Selamat Pagi", dapat memiliki berbagai bentuk, tergantung bahasanya:

POLIMORFISME

Singkatnya Polimorfisme adalah suatu perilaku (method) yang sama dapat memiliki banyak bentuk Misalnya "Selamat Pagi", dapat memiliki berbagai bentuk, tergantung bahasanya:

- Selamat pagi! (Indonesia)
- Good morning! (Inggris)
- Buenos días! (Spanyol)
- Bonjour! (Perancis)
- Guten Morgen! (Jerman)
- Buongiorno! (Italia)
- おはようございます (Ohayou gozaimasu)! (Jepang)
- 안녕하세요 (Annyeonghaseyo)! (Korea)

POLIMORFISME DI PHP

Pada sebagian besar bahasa pemrograman, termasuk PHP, polimorfisme diimplementasikan dengan **Abstract Class** atau **Interface**

```php
abstract class Orang {
  abstract public function selamatPagi();
}

class OrangIndonesia extends Orang {
  public function selamatPagi() {}
}
```

```php
interface Orang {
  public function selamatPagi();
}

class OrangIndonesia implements Orang {
  public function selamatPagi() {}
}
```

POLIMORFISME DI PHP

Abstract Class

```php
abstract class Orang {
  abstract public function etnis();
  abstract public function selamatPagi();
}

class OrangIndonesia extends Orang {
  public function etnis() {
    return "Indonesia";
  }

  public function selamatPagi() {
    return "Selamat Pagi !";
  }
}
```

POLIMORFISME DI PHP

Interface

```php
interface Orang {
  public function etnis();
  public function selamatPagi();
}

class OrangIndonesia implements Orang {
  public function etnis() {
    return "Indonesia";
  }

  public function selamatPagi() {
    return "Selamat Pagi !";
  }
}
```